

Quando um bloco começa a ser executado, as variáveis declaradas dentro dele são ativadas, tornando-se automaticamente *ativas*, e permanecem ativas até o término de execução do bloco. Em outros palavras, uma variável declarada no início de um bloco só existe dentro dele. Diz-se então que as variáveis são *locais* ao bloco, o que permite que elas só ocupem memória quando necessário.

Com isto, o programador pode gerenciar a utilização da memória. Por exemplo, um vetor cujas dimensões não se conhecem a priori, não precisa ser lida ou depende de algum procedimento, pode ser declarado dentro de um bloco e, portanto, alocado dinamicamente na memória.

Exemplo:

```

início
  inteiro N, i
  leia (N)
  início
    vetor v = vetor[1..N] real
    i ← 1
    para i de 1 até N faça
      leia (v[i])
    fim para
    imprime (v)
  fim
fim

```

7.2 ESCOPO DE VARIÁVEIS

Diz-se que um bloco é *externo* a outro, quando o segundo faz parte do primeiro.

Assim sendo, uma variável declarada em um bloco é *global* para todos os blocos internos e *local* para o próprio bloco.

Podemos ter diversos blocos aninhados, conforme mostra a figura a seguir.



nível	bloco
0	M
1	P, Q
2	A, R, S
3	B

i e *j* são internos a *M*
A é interno a *P*
B é interno a *A*
R e *S* são internos a *Q*

M é externo a *P* e *Q*
P é externo a *A*
A é externo a *B*
Q é externo a *R* e *S*

Conforme foi mostrado, uma variável declarada dentro de um bloco só é conhecida dentro deste bloco.

Se uma variável *A* declarada em um bloco já foi declarada com mesmo nome num bloco mais externo, a variável ativa no bloco é aquela declarada localmente. A variável *A* deixa de ser global para aquele bloco.

Os exemplos a seguir ilustram variáveis locais, globais e escopo de variáveis.

Exemplo

```

início
  inteiro i, j
  leia (i)
  i ← i + 2
  início
    real X
    X ← j + 1
    imprime (X)
  fim
  leia (j)
  i ← i + j
fim

```



No exemplo acima, *i* e *j* são variáveis globais e *X* é uma variável local ao bloco interno.

```

início
  inteiro i, j
  leia (i)
  i ← i + 2
  início
    real X, i
    X ← j + 1
    imprime (X)
  fim
  leia (j)
  i ← i + j
fim

```



No exemplo acima, *i* deixa de ser global por ter sido declarada (com o mesmo nome) no bloco interno. Na verdade, é como se fosse uma nova variável *i*.